

法政大学学術機関リポジトリ  
HOSEI UNIVERSITY REPOSITORY

# 分散環境における超並列計算に対する計算限界の解明とその計算パラダイムの創出

著者	和田 幸一
ページ	1-5
発行年	2017-06-01
URL	<a href="http://hdl.handle.net/10114/00021468">http://hdl.handle.net/10114/00021468</a>

平成 29 年 6 月 1 日現在

機関番号：32675

研究種目：基盤研究(C) (一般)

研究期間：2014～2016

課題番号：26330020

研究課題名(和文)分散環境における超並列計算に対する計算限界の解明とその計算パラダイムの創出

研究課題名(英文)Limitations of Massively Parallel Computation on Distributed Environment

研究代表者

和田 幸一 (WADA, Koichi)

法政大学・理工学部・教授

研究者番号：90167198

交付決定額(研究期間全体)：(直接経費) 3,700,000 円

研究成果の概要(和文)：本研究は、ビッグデータ処理に適した新しい超大規模並列計算モデルの創出を目指し、MapReduce計算における理論モデルを確立し、その計算限界を解明することを目的としている。(1)従来のモデルと並列アルゴリズムの整理と検討、(2)MapReduce理論モデルの確立とその基本演算の実現、(3)MapReduce計算に対する計算限界の解明、(4)ビッグデータの特徴と耐故障計算とモデルとの関係の解明などを行った。

研究成果の概要(英文)：In this research in order to create new massively parallel computational models for Bigdata Processing, we establish a theoretical model for MapReduce Computation and figure out limitation of its computation. We have mainly done the following research; (1) examination of relationship between previous models and our model, (2) establishment of a theoretical model of MapReduce computational, (3) Clarification of limitation of its computational power, and (4) examination of relationship of Bigdata processing and fault-tolerant computations.

研究分野：並列分散アルゴリズム

キーワード：MapReduce計算 並列複雑度 組合せ回路 PRAM 計算限界

## 1. 研究開始当初の背景

並列計算の究極の理論モデルは 1970 年代に提案された PRAM(Parallel Random Access Machine) である。PRAM は共有メモリ型並列計算機のモデルで、問題が内包する本質的な並列性を理論的に解明するために導入された。並列計算機モデルの提案以降、CPU の高速化とメモリや通信コストの低下により、並列計算機が現実設計されるようになり、並列計算機に適したさまざまなアーキテクチャや手法が考案された。理論と実際の協調的発展により並列計算の夜明けはすぐそこに迫っているように見えた。しかしながら、PRAM は究極の並列計算機モデルであり、現実の並列計算機での効率的な実現は容易でなく、理論と現実が乖離するようになり、より現実的な並列計算理論モデルも提案されたが、以降並列計算機の理論も実際も衰退の途をたどる。「処理、メモリ、通信に要するコストの急速な低下により、演算集約型の領域においては、少なくともこの 20 年のうちに、逐次マシンが並列マシンによって最終的に置き換えられることが不可避であることが明らかになってきた。しかし、この現象はおきていない。」という逐次マシンから並列マシンへの移行が不可避にもかかわらず、その現象は起きていないという Leslie Valiant の言葉のとおり、並列計算の夜明けはすぐそこにはなく、並列計算機はマルチスレッドを備えた単一計算機にとって代われ、並列計算は理論的にも実際的にも衰退の時代を迎えた。並列処理は特殊な要求を除き、マルチスレッドの単一計算機で十分であると思われていた。また、高速なクロックや深いパイプライン、スーパースカラーアーキテクチャも投資に見合うだけの成果は生まれてこなかったこともそれを裏付けている。しかしながら、ここにきてペタ、エクサ級のビッグデータを取り扱うためには並列処理は不可避であり、新しい戦略が必要である。

## 2. 研究の目的

新しい大規模並列計算として、ビッグデータに対する分散処理を表現するためのプログラミングモデルである MapReduce が注目されている。MapReduce は Google で開発され、ビッグデータ解析のデファクトスタンダードになりつつあり、Hadoop によるオープンソース実装も相まって Yahoo, Facebook, Adobe, IBM などでも実際に使用されている。また、現時点では、MapReduce に対しては、いくつかの理論モデルが提案されているが、理論モデルはまだ固まっていない。MapReduce はこのモデル化のもとで、ある種の PRAM アルゴリズムを効率よくシミュレートできる可能性を秘めているため、本研究では、まず、MapReduce を代表とする超大規模計算の理論モデルを確立し、そのうえで並列計算としての計算限界を解明する。

MapReduce における並列化は、PRAM な

どの従来のものとは全く異なるため、これまで提案されている並列計算のパラダイムは適用できないものが多い。本研究では新しい並列計算のアルゴリズム設計理論の開発を目指す。それをさらに押し進めて、ビッグデータ処理を踏まえた新しい超大規模並列計算モデルを創出するのが、本研究の最終目的である。

本研究の独創的な点は、従来とは異なる並列のとらえ方を理論に取り込むことにある。MapReduce の並列性はこれまでの並列計算 (PRAM) のそれとは全く異なっている。

MapReduce ではデータをキーと値の組の並びとし、Map 操作によって、キーと値の組に新しいキーをつけることによって、同じキーをもつデータごとに分類し（この操作は shuffle と呼ばれる）、集められた同じキーをもつデータを Reduce 操作で処理をする。これを一連の操作（ラウンドと呼ばれる）として、Reduce 操作の結果を次の Map 操作の入力とすることを繰り返すことによって計算が行われる。Map 操作と Reduce 操作が逐次でストリーム計算が行えるように、すなわち、それぞれのデータに対して、すべての入力データを見なくても処理が可能であるようにしさえすればよい。この Map 操作と Reduce 操作の部分が並列に実行可能になるのであるが、PRAM などのように並列に実行可能な部分を見つけ出す必要はない。従って、MapReduce の基本的なアルゴリズムの設計手法としては、問題の並列性を抽出するかわりに、ラウンド数が少なくなるように、MapReduce の繰り返しに変換すればよい。

もともと、MapReduce はビッグデータ解析を目的として開発されたものであるから、MapReduce 操作において、ストリーム計算をしなければならないのであるが、ここではそれを並列計算の設計パラダイムと考える。また、逆に問題の並列性が抽出できていれば、それをストリーム計算に落とすことは難しくなく、よって、以下のような新しい視点が可能になる。

(1) これまで並列化が難しいとされていた問題、すなわち、PRAM では対数多項式時間では解けない問題が MapReduce では解ける可能性がある。

(2) PRAM で並列化可能な問題は、PRAM の 1 ステップを MapReduce の 1 ラウンドで模倣できるので、PRAM での計算時間と同じラウンド数で MapReduce によって解くことはできるが、MapReduce では PRAM の計算時間よりも漸近的に少ないラウンド数で実現できる可能性がある。

(3) これまでの並列計算において、最速ではない並列アルゴリズム（例えば、逐次アルゴリズムで  $O(n^3)$  時間を並列で  $O(n)$  時間にしたもの）を MapReduce へ変換することができれば、高速化することが可能になり、これまでの遺産が再び脚光を浴びる可能性がある。

MapReduce に対する並列計算モデルの構

築とその計算限界の解明は、以上のような新しい問題の解決に寄与できる。並列計算量理論的には、MapReduce によって対数多項式で計算できる問題のクラス(MRC と呼ぶ)と NC や P の中で並列化できそうにない問題のクラス P-完全との関係を明らかにすれば、これまで未解決であった  $P \neq NC$  の糸口になるかもしれない。上記の考察から明らかに MRC は NC を含み、P-完全問題を包含する可能性もある。

また本研究は、MapReduce の計算限界を明らかにするだけにとどまらず、これらの計算限界の解明を通して、MapReduce に代わる新しい並列計算モデルの創出にチャレンジするものである。「数が質を本質的に変える」という現象が VLSI 回路設計では起こった。回路の VLSI 化に際しては、設計手法も評価尺度も本質的に変わったわけであるが、ビッグデータの場合は、数の増大は VLSI の場合に比べてさらに大きくなるので、本質的に変化するものが何であるかを明らかにしなければならない。これこそが本研究の最終目的である。

### 3. 研究の方法

ビッグデータ処理に適した新しい超大規模並列計算モデルの創出を念頭に置きながら、MapReduce の理論モデルの確立と計算限界の解明を行う。具体的には以下のとおりである。

(1) (従来のモデルと並列アルゴリズムの整理と検討) 従来から提案されている PRAM の後継者を整理する。これらは PRAM の現実的でない部分をより現実的に取り扱いが可能なものにしたものである。また、これらの上で開発された並列アルゴリズムを整理する。並列アルゴリズムに関しては、PRAM 上のものだけではなく、回路で実現されたものや分散網で実現されたものを含めて考える。これらの検討は、PRAM 型のモデルから新しいモデルに継承できる部分とそうでない部分を明らかにすることを目的とする。並列アルゴリズムに関しても、MapReduce にそのまま変換可能なもの、そのままでは変換できないがすこし変形すれば変換可能なもの、まったく変換できないものを分類し、MapReduce に適したアルゴリズム手法を検討する。また、従来並列アルゴリズム設計にはまったく適さないが、MapReduce には向いている設計手法の洗い出しを行う。また、これまでに開発された MapReduce アルゴリズムも整理検討し、MapReduce 向きアルゴリズム設計手法を確立する。

(2) (モデルの確立と基本演算の実現) (1)の検討を基に MapReduce の理論モデルを設計する。これまでいくつかの理論モデルが存在するが、それらも検討したうえで、最終目的であるビッグデータ処理に適した新しい超大規模並列計算モデルの創出を考慮しつつ、現時点で最も MapReduce に適したモデルを構築する。そして、ソーティングや検索等の基本

演算や並列処理に必要な基本演算(リストランキング、プレフィックス計算、彩色)を実現する。

### (3) (MapReduce の計算限界の解明)

MapReduce の並列計算モデルを構築する上で、必須事項は Map 操作と Reduce 操作においてはすべての入力データを蓄えることができず、ストリーム計算になることと、Mapper (Map 操作を実行する実体) と Reducer (Reduce 操作を実行する実体) のメモリ量は全体の入力サイズ ( $n$  とする) の劣線形 ( $o(n)$ ) でなければならないことである。この制限によってすべての入力を一つの Reducer に集めて処理することができないので、P のすべての問題を MapReduce で計算できるかも自明ではない。前項で述べたように、このように制限をしても、NC のすべての問題は MapReduce で計算可能である。すなわち、これまで並列化が可能であるすべての問題は MapReduce によって同程度の時間で計算できる。この2つがスタート地点である。ここから先は、モデルをいかに設定するか依存するので、MapReduce の計算限界は如何に現実の MapReduce をうまくとらえることができるかにかかっている。また、これまでに提案された MapReduce のモデルでは、Map 操作と Reduce 操作はストリーム計算であるので、計算時間はせいぜい入力数の線形程度であり、考慮しなくてもよいという近似モデルである。これが実際に成り立つのか、あるいはこれらの時間は無視できないようになるかも明らかにしなければならない。現時点では、Map 操作や Reduce 操作にかかる時間は全体の時間に対して無視できなくなると予想しており、それらを考慮しても PRAM における対数多項式時間を凌駕できるような MapReduce アルゴリズムを開発しなければならない。これが可能ならば、MapReduce に適した設計手法につながることを期待できる。

(4) (ビッグデータの特徴や耐故障性計算とモデルとの関係) ビッグデータの特徴である 3V (Volume:データ量, Variety:多様性, Velocity:データの生成頻度) が MapReduce に対する並列計算モデルにどのように影響するかを明らかにする。これまでの MapReduce のモデル化においては、データ量以外のものは考慮されていない。理論モデルには他のものは必要ないのかどうかを検討する。

### 4. 研究成果

ビッグデータ処理に適した新しい超大規模並列計算モデルの創出を念頭に置きながら、MapReduce の理論モデルを確立し、そのうえで計算限界をいくつか解明した。詳細は発表論文に譲るが、主要な結果は以下の通りである。

組合せ回路で効率よく計算できる問題のクラスと MapReduce モデルにおいて定数ラウンドで実現できる問題のクラスを検討した。

従来は、正規言語を含む  $o(\log n)$  領域で解ける問題のクラスは MapReduce において定数ラウンドで解けることが分かっていたが、このクラスを真に拡張し、ファンイン無制限で定数段数の多項式サイズ組合せ回路で計算できる問題のクラスが定数ラウンドで実現できることを示した。ここで用いた方法は汎用性があり、さらにこれを拡張し、多項式サイズの段数  $O(\log n)$  の組合せ回路で実現できる問題のクラス ( $NC^1$ ) が MapReduce モデルにおいて定数ラウンドで実現できることを示した。

並列計算機の古典理論モデルである PRAM (Parallel Random Access Machine) において、これまで未解決であった EREW-PRAM と CREW-PRAM の関係を明らかにした。CREW-PRAM において、 $T$  時間の並列アルゴリズムで解ける問題は EREW-PRAM では  $O(T \log n)$  時間で解けることは分かっていたが、EREW-PRAM において  $O(T + \log n)$  時間で解けるための十分条件を与えた。

これまでに示した MapReduce を分散環境で実現する場合の基礎的な考察を進めて、分散計算における計算能力の限界を明らかにし、分散計算における新たな故障を考慮した場合の計算能力や自己安定性に関する結果も導出した。

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

〔雑誌論文〕(計 6 件)

- ① 三田恵利佳：並列計算機理論モデル PRAM における計算能力に関する研究，法政大学理工学部応用情報工学科 2016 年度卒業論文，1-24 (2017-03)。(査読なし)
- ② D. Canepa, X. Defago, T. Izumi, M. Potop-Butucaru : Flocking with oblivious robots, Stabilization, Safety, and Security of Distributed Systems, LNCS, 10083, 94-108 (2016)。(査読あり)
- ③ 間々田剛史：MapReduce 計算の並列複雑さに関する研究，法政大学大学院理工学研究科応用情報工学専攻 2015 年度修士論文，1-25 (2016-03)。(査読なし)
- ④ F. Bonnet, X. Defago, T. D. Nguyen, M. Potop-Butucaru : Tight bound on mobile Byzantine agreement, Theoretical Computer Science, 609, 361-373 (2016)。(査読あり)
- ⑤ Ta. Izumi, To. Izumi, H. Ono, K. Wada : Approximability of minimum certificate dispersal with tree structures, Theoretical Computer Science, 591, 5-14 (2015)。(査読あり)
- ⑥ To. Izumi, K. Kinpara, Ta. Izumi, K. Wada : Space-efficient self-stabilizing counting population protocols on mobile sensor networks, Theoretical Computer Science, 552, 99-108 (2014)。(査読あり)

〔学会発表〕(計 8 件)

- ① K. Wada : Computational power of autonomous mobile robots with lights - boundary between solvability and unsolvability, The 4<sup>th</sup> International Symposium on Computing and Networking (CANDAR2016), Hiroshima, Japan (2016. 11. 22-25) (招待講演)。
- ② X. Defago : Intrusion-tolerance and dynamic faults in the internet of things, AEARU Web Technology and Computer Science Workshop 2016 (WTCS2016), Sendai, Japan (2016. 9. 20-21)。
- ③ X. Defago : Intrusion-tolerance and dynamic faults in Distributed Systems, Japan-Taiwan Workshop on Secure and Dependable IoT Systems, Tokyo, Japan (2016. 7. 24)。
- ④ K. Wada : Parallel complexity on MapReduce computation, 1<sup>st</sup> International Conference on Advanced Information and Communication Technology 2016 (ICAICT 2016), Chittagong, Bangladesh (2016. 5. 16-17) (招待講演)。
- ⑤ K. Wada, Y. Katayama, S. Terai : A new model of mobile robots with lights and its computational power, 5<sup>th</sup> Workshop on Moving And Computing (MAC2015), Montreal, Canada (2015. 10. 19-21)。
- ⑥ S. Terai, K. Wada, Y. Katayama : A new model for autonomous mobile robots with lights and its solvability of gathering problems, Workshop on Distributed Robotic Swarms (WDRS 2015), Tokyo, Japan (2015. 10. 5)。
- ⑦ A. Maurer, S. Tixeuil, X. Defago : Communicating reliability in multihop dynamic networks despite Byzantine failures, SRDS 2015, Montreal, Canada (2015. 9. 28-10. 1)。
- ⑧ K. Wada : Computational models for big data processing, 6<sup>th</sup> International Workshop on Parallel and Distributed Algorithms and Applications, Shizuoka, Japan (2014. 12. 10-11) (招待講演)。

〔図書〕(計 3 件)

- ① X. Defago : Atomic Broadcast, in Encyclopedia of Algorithms, Ed. Ming-Yang Kao, Springer, 2389(155-160) (2016)。
- ② X. Defago : Causal order, logical clocks, state machine replication, in Encyclopedia of Algorithms, Ed. Ming-Yang Kao, Springer, 2389(283-286) (2016)。
- ③ 離散数学，陳慰，和田幸一共著，森北書店，168(1-168) (2014-11)。

## 6. 研究組織

### (1) 研究代表者

和田 幸一(WADA, Koichi)  
法政大学・理工学部・教授  
研究者番号：90167198

### (2) 研究分担者

デファゴ クサヴィエ(DEFAGO, Xavier)  
東京工業大学・情報理工学院・教授  
研究者番号：7033557